

CLAIM AMENDMENTS

5 *sub 1*
1 (CURRENTLY AMENDED): A system comprising:
a directory of identifiers and metadata to a plurality of network services;
an engine for receiving requests, ~~and~~ using said identifiers in said directory to direct said requests to access said network services when requested, and constructing a state storing session for interfacing with said network services, wherein said session uses a driver to interface with each of said network services and said session is configured from said metadata from said directory; and
10 ~~a plurality of drivers for interfacing with said plurality of network services and with said engine; and~~
a plurality of service providers accessible to said plurality of drivers for providing network services identified in said directory.

15 2 (CANCELLED)

3 (CURRENTLY AMENDED): The system of claim 2 1 wherein said metadata defines a schema of a network service's input and output interfaces.

20 4 (PREVIOUSLY AMENDED): The system of claim 3 wherein said metadata further includes configuration parameters for configuring a driver associated with said network service.

5 (PREVIOUSLY AMENDED): The system of claim 1 wherein said network services are accessible via an API.

6 (PREVIOUSLY AMENDED): The system of claim 1 wherein said network services are XML based network services.

5 7 (PREVIOUSLY AMENDED): The system of claim 1 wherein said a network service provider comprises an entity that is capable of receiving some information and providing a response.

8 (PREVIOUSLY AMENDED): The system of claim 1 wherein said engine interprets
10 said requests and determines what network services are needed, directs requests to the appropriate network services via said network service drivers, and builds responses into replies.

9 (PREVIOUSLY AMENDED): The system of claim 1 wherein said requests comprise HTTP requests.

10 (CANCELED)

11 (CURRENTLY AMENDED): A method for accessing network services comprising:
storing identifiers and metadata of a plurality of network services in a directory;
20 providing requests to an engine wherein said engine uses said identifiers to direct said requests to access said plurality of network services when requested; and
constructing a session for interfacing with said network service, wherein said session uses a driver to interface with said network service and said session is configured from said metadata from said directory;

interfacing with said plurality of network services and with said engine via a plurality of drivers based on said requests.

12 (CANCELLED)

5

13 (CURRENTLY AMENDED): The method of claim ~~12~~ 11 wherein said metadata defines a schema of a service's input and output interfaces.

14 (PREVIOUSLY AMENDED): The method of claim 13 wherein said metadata further includes configuration parameters for configuring a driver associated with said network service.

10

15 (PREVIOUSLY AMENDED): The method of claim 11 wherein said network services are accessible via an API.

15

16 (PREVIOUSLY AMENDED): The method of claim 11 wherein said network services are XML based network services.

20

17 (PREVIOUSLY AMENDED): The method of claim 11 wherein said a network service provider comprises an entity that is capable of receiving some information a providing a response.

18 (PREVIOUSLY AMENDED): The method of claim 11 wherein said engine interprets said requests and determines what network services are needed, directs request to the appropriate network services via said service drivers, and builds responses into replies.

19 (PREVIOUSLY AMENDED): The method of claim 11 wherein said requests comprise HTTP requests.

5 20 (CANCELED)

C 21 (NEW): A method for accessing network services comprising:
storing identifiers of a plurality of network services in a directory, wherein said identifiers comprise of input and output schemas for each of said plurality of network services;
10 providing a request to an engine, wherein said engine searches said identifiers in said directory to direct said request to an appropriate network service;
constructing a session for interfacing with said network service, wherein said session uses a driver to interface with said network service and said session is configured from said metadata from said directory;
15 sending input data to said network service according to said input schema; and
receiving response data from said network service according to said output schema.

22 (NEW): The method of claim 21 further comprises:
sending login data to said network service for authentication; and
20 receiving authentication from said network service before granting access to said network service.

23 (NEW): The method of claim 22 wherein said login data is in sent via a packet of XML.

24 (NEW): The method of claim 22 further comprises:

caching said authentication received from said network service in said session.

5 25 (NEW): The method of claim 21 wherein said step of sending input data further comprising:

initializing an instance of said driver associated with said network service; and
calling an execute method in said driver.

10 26 (NEW): The method of claim 25, further comprising:
validating said input data.

27 (NEW): The method of claim 21 wherein said step of receiving response data further comprises:

15 receiving said response data from said network service;
formulating said response data received from said network service in an appropriate
format; and
20 sending said formulated data back in response to said request.

28 (NEW): The method of claim 27 wherein said format is HTTP.

29 (NEW): The method of claim 21 wherein said input data is in XML.

30 (NEW): The method of claim 21 wherein said output data is in XML.

31 (NEW): A method for using a web service, comprising:
specifying metadata for a web service;
identifying a service driver class for a service driver associated with said web service,
5 wherein said service driver class is configured for use in a web services engine;
supplying a plurality of web service parameters that configure said service driver; and
defining an XML input and output interfaces for said web service; and
using a directory to store said metadata and said input and output interfaces for said web
service.

10 32 (NEW): The method of claim 31 wherein said service driver class is implemented in
Java.

15 33 (NEW): The method of claim 31 wherein said web service parameters are supplied by
said web service to said web service driver upon invocation of said web service in said web
services engine.

20 34 (NEW): The method of claim 33 wherein said web service parameters are blocks of
XML data in a pre-defined schema.

35 (NEW): The method of claim 31 wherein said XML input and output interfaces
conform to a pre-defined schema, wherein inputs and outputs are defined as lists of XML
structured parameters from a set of supported data types in said schema.

36 (NEW): The method of claim 31 wherein said XML input and output interfaces are defined by an external URL that represents an XML document that follows a standard schema.

37 (NEW): The method of claim 33 wherein said standard schema is DTD.

38 (NEW): The method of claim 31 wherein said directory is implemented using the LDAP protocol.

39 (NEW): The method of claim 31 wherein said web service driver is implemented in Java.

40 (NEW): The method of claim 31 wherein said web service driver invokes other web service via the API of said web services engine.

41 (NEW): The method of claim 31 wherein said web service driver converts an input in the form of XML to a format of input accepted by an external application.

42 (NEW): The method of claim 31 wherein said web service driver converts a format of output given by an external application to an output in the form of XML.

43 (NEW): The method of claim 31 wherein said web service driver converts an input in the form of XML to a HTTP format of input accepted by a web application.

44 (NEW): The method of claim 31 wherein said web service driver converts a HTTP format of output given by a web application to an output in the form of XML.

45 (NEW): The method of claim 31 wherein said web services engine performs a plurality of web services in the context on a single session.

46 (NEW): The method of claim 31 further comprising:
caching runtime models in said web services engine; and
customizing said runtime models at run time to transform them into web services application (WSA).

47 (NEW): The method of claim 46 wherein said runtime model comprises:
definitions for a plurality of actions;
definitions for a plurality of parameters used as web service inputs and outputs;
definitions for constructing user interface pages.

48 (NEW): The method of claim 47 wherein said user interface pages comprise:
user interface elements that generate subsequent requests to said WSA.

49 (NEW): The method of claim 46 wherein said WSA can execute multiple web services in one running session.

50 (NEW): A method of invoking a web services application (WSA), comprising:
accepting a request to run a model based service;

loading an instance of model runner in response to said request;
loading a runtime model in said instance of model runner; and
generating a WSA based on said loaded runtime model.

5 51 (NEW): The method of claim 50 wherein said request is generated by an user interface element on a web page.

52 (NEW): The method of claim 50 wherein said model based service maintains a plurality of runtime models that correspond a plurality of WSAs, wherein functionality of each
10 of said WSAs is defined by one of said runtime models.

53 (NEW): The method of claim 50 wherein said instance of model runner uses said runtime model as a service parameter.

15 54 (NEW): The method of claim 50 further comprises:
executing the WSA; and
saving the state of WSA in a session.

55 (NEW): The method of claim 54 further comprises:
20 accepting a new request from a user interface page generated by a prior request, whereby said new request uses a handle to said session to access said WSA.

56 (NEW): The method of claim 55 further comprises:
starting a second instance of model runner and connecting to said WSA;

executing trigger named in said new request; and

sending a response in reply to said new request.

57 (NEW): The method of claim 55 wherein said user interface page is in XML.

58 (NEW): The method of claim 52 wherein said new request identifies an action to be executed in said WSA.

59 (NEW): The method of claim 50 wherein said WSA is represented as a plurality of XML parameters.

60 (NEW): The method of claim 59 wherein said XML parameters comprise data that can be modified over time.

61 (NEW): The method of claim 50 wherein said WSA can generate additional service calls to a plurality of web services.

62 (NEW): The method of claim 50 wherein said runtime model further comprises:

a Model root node, wherein said Model root node further comprises:

a plurality of page entities;

a plurality of service call entities;

a plurality of trigger entities;

a plurality of parameter entities; and

a plurality of behavior entities.

63 (NEW): A system for providing network service comprising:

a directory of identifiers and metadata to a plurality of network services;

an engine for receiving requests, using said identifiers in said directory to direct said requests to access said network services when requested, and constructing a state storing session for interfacing with said network services, wherein said session uses a driver to interface with each of said network services and said session is configured from said metadata from said directory;

a plurality of service providers accessible to said plurality of drivers for providing network services identified in said directory, wherein input data is sent to said network service according to an input schema in said metadata and response data is received from said network service according to an output schema in said metadata.

64 (NEW): The system of claim 63 further comprises:

a plurality of service parameters that configure said driver; and
an XML input and output interfaces for said network service; and

65 (NEW): The system of claim 63 wherein login data is sent to said network service for authentication and an authentication is received from said network service before access is granted to said network service.

66 (NEW): The system of claim 65 wherein said login data is in sent via a packet of XML.

67 (NEW): The system of claim 65 wherein said authentication received from said network service in said session is cached for later use within said session.

68 (NEW): The system of claim 63 wherein said input data is sent after an instance of said driver associated with said network service is initialized and an execute method in said driver is called.

69 (NEW): The system of claim 63 wherein said input data is validated.

70 (NEW): The system of claim 63 wherein said response data received from said network service is formulated into an appropriate format and sent back to an entity that originates said request.

71 (NEW): The system of claim 70 wherein said format is HTTP.

72 (NEW): The system of claim 63 wherein said input data is in XML.

73 (NEW): The system of claim 63 wherein said output data is in XML.

74 (NEW): A system for using a web service, comprising:
a web services engine for receiving requests;
metadata specified for said web service;
a service driver class for a service driver associated with said web service, wherein said service driver class is configured for use in said web services engine;

a plurality of web service parameters that configure said service driver; and
an XML input and output interfaces for said web service; and
a directory to store said metadata and said input and output interfaces for said web
service.

5

75 (NEW): The system of claim 74 wherein said service driver class is implemented in
Java.

10

76 (NEW): The system of claim 74 wherein said web service parameters are supplied by
said web service to said web service driver upon invocation of said web service in said web
services engine.

15

77 (NEW): The system of claim 76 wherein said web service parameters are blocks of
XML data in a pre-defined schema.

20

78 (NEW) The system of claim 74 wherein said XML input and output interfaces
conform to a pre-defined schema, wherein inputs and outputs are defined as lists of XML
structured parameters from a set of supported data types in said schema.

79 (NEW): The method of claim 74 wherein said XML input and output interfaces are
defined by an external URL that represents an XML document that follows a standard schema.

80 (NEW): The system of claim 76 wherein said standard schema is DTD.

81 (NEW): The system of claim 74 wherein said web services directory is implemented using the LDAP protocol.

5 82 (NEW): The system of claim 81 wherein said web service driver is implemented in Java.

83 (NEW): The system of claim 74 wherein said web service driver invokes other web service via the API of said web services engine.

10 84 (NEW): The system of claim 74 wherein said web service driver converts an input in the form of XML to a format of input accepted by an external application.

15 85 (NEW): The system of claim 74 wherein said web service driver converts to a format of output given by an external application to an output in the form of XML.

86 (NEW): The system of claim 74 wherein said web service driver converts an input in the form of XML to a HTTP format of input accepted by a web application.

20 87 (NEW): The system of claim 74 wherein said web service driver converts to a HTTP format of output given by a web application to an output in the form of XML.

88 (NEW): The system of claim 74 wherein said web services engine performs a plurality of web services in the context on a single session.

89 (NEW): The system of claim 74 further comprising:

runtime models cached in said web services engine, wherein said runtime models are customized at run time to be transformed into web services application (WSA).

5

90 (NEW): The system of claim 89 wherein said runtime model comprises:

definitions for a plurality of actions;

definitions for a plurality of parameters used as web service inputs and outputs;

definitions for constructing user interface pages.

10

91 (NEW): The system of claim 90 wherein said user interface pages comprise:

user interface elements that generate subsequent requests to said WSA.

92 (NEW): The system of claim 90 wherein said WSA can execute multiple web services

15 in one running session.

93 (NEW): A system for invoking a web services application (WSA), comprising:

an engine for accepting a request to run a model based service;

an instance of model runner to be loaded by said engine in response to said request;

20

a runtime model to be loaded in said instance of model runner; and

a WSA to be generated based on said loaded runtime model.

94 (NEW): The system of claim 93 wherein said request is generated by an user interface element on a web page.

95 (NEW): The system of claim 93 wherein said model based service maintains a plurality of runtime models that correspond a plurality of WSAs, wherein functionality of each of said WSAs is defined by one of said runtime models.

5

96 (NEW): The system of claim 93 wherein said instance of model runner uses said runtime model as a service parameter.

97 (NEW): The system of claim 93 wherein said WSA is executed and the state of said WSA is saved in a session.

10

98 (NEW): The system of claim 97 wherein a new request from a user interface page is generated by a prior request, whereby said new request uses a handle to said session to access said WSA.

15

99 (NEW): The system of claim 98 further comprises:
a second instance of model runner for connecting to said WSA in response to said request, wherein a trigger named in said new request is executed and a response is sent in reply to said new request.

20

100 (NEW): The system of claim 99 wherein said user interface page is in XML.

101 (NEW): The system of claim 95 wherein said new request identifies an action to be executed in said WSA.

102 (NEW): The system of claim 93 wherein said WSA is represented as a plurality of XML parameters.

5 103 (NEW): The system of claim 102 wherein said XML parameters comprise data that can be modified over time.

10 104 (NEW): The system of claim 93 wherein said WSA can generate additional service calls to a plurality of web services.

105 (NEW): The system of claim 93 wherein said runtime model further comprises:
a Model root node, wherein said Model root node further comprises:

a plurality of page entities;

a plurality of service call entities;

15 a plurality of trigger entities;

a plurality of parameter entities; and

a plurality of behavior entities.